CALCULADORAS PROGRAMABLES DE BOLSILLO



NUEVAS TECNOLOGIAS

BIBLIOTECA DE ELECTRONICA/INFORMATICA





BIBLIOTECA DE ELECTRONICA/INFORMATICA

CALCULADORAS PROGRAMABLES DE BOLSILLO



Esta obra es una nueva edición actualizada y ampliada de la obra originalmente publicada por Marcombo, S.A. de Boixareu editores, con el título de «Aplicaciones de la Electrónica»

El contenido de la presente obra ha sido realizado por Marcombo, S.A. de Boixareu editores, bajo la dirección técnica de José Mompin Poblet, director de la revista «Mundo Electrónico»

ISBN 84-7634-485-6 (Obra completa)
ISBN 84-7634-852-5 (Vol. 50)
D. L.: B. 39443-1986

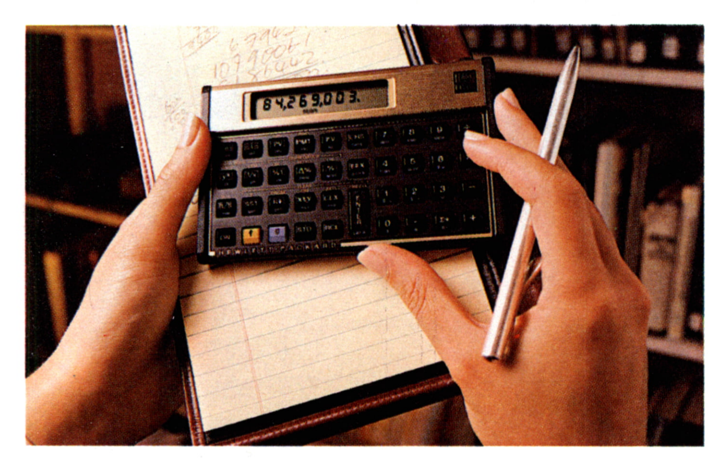
Impreso y encuadernado por
printer industria gráfica, sa c.n. II, cuatro caminos, s/n
08620 sant vicenç dels horts barcelona 1986

Printed in Spain

Calculadoras programables de bolsillo

INTRODUCCION

Las calculadoras programables de bolsillo aparecieron cuando el tratamiento de la información se realizaba con grandes computadores. Entonces no existían los *computadores personales* que actualmente compiten en precio con las pequeñas calculadoras programables.



El diseño de la calculadora recuerda la forma de la regla de cálculo, su predecesora en la obtención de operaciones matemáticas. (Cortesía: Hewlett Packard).

Como estos grandes computadores no se podían transportar fue necesario reducir su tamaño, y con ello sus capacidades, para utilizar la gran capacidad de cálculo de la electrónica digital, que por otra parte fue miniaturizándose poco a poco. Esta miniaturización hizo posible que la electrónica sustituyese a las reglas de cálculo logarítmicas en sus aplicaciones técnicas comerciales y clásicas para los estudiantes.

En efecto: la regla de cálculo rindió un gran servicio por la

rapidez en obtener unas tres cifras significativas de los complicados cálculos, pero era necesario afinar la vista para obtener exactitud en la coincidencia de las marcas y recordar las veces que la reglilla salía por un lado o por el otro para estar seguro en la posición de la coma decimal. Por ello los constructores de calculadoras de bolsillo comprendieron el gran mercado que podían tener entre los habituales usuarios de las reglas de cálculo, con la ventaja de que los cálculos podían ser programables.

Primero se construyeron calculadoras no programables, pero el hecho de utilizar las fórmulas de manera repetida hizo que se diseñasen las programables, primero sin la posibilidad de bifurcar el programa y luego con toda clase de comandos y funciones numéricas y alfanuméricas, con periféricos o sin ellos, es decir, solamente los computadores personales han frenado su diversificación por el momento.

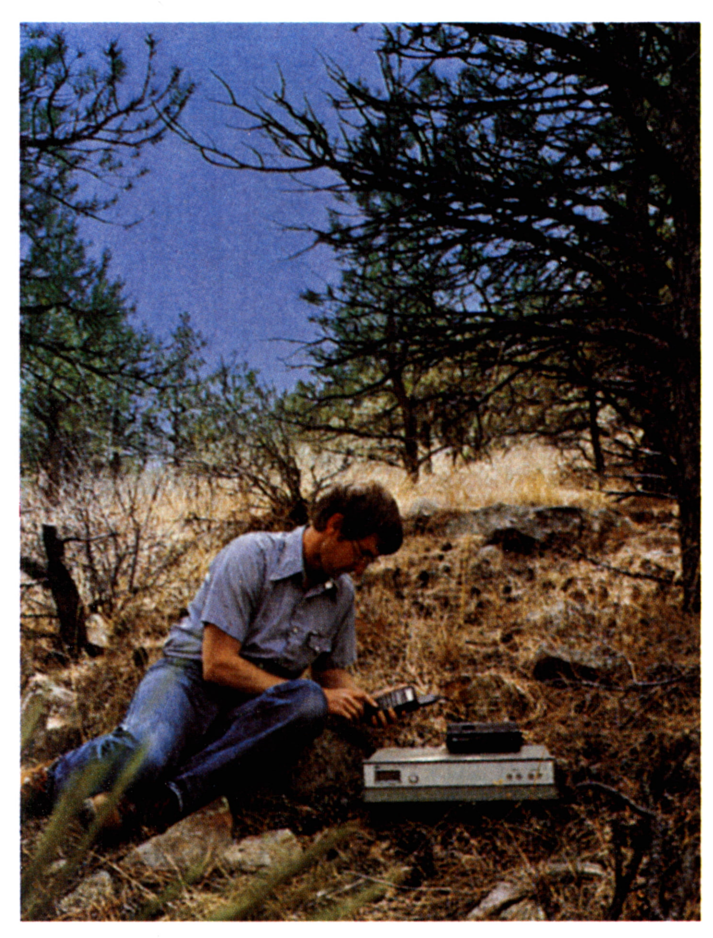


Calculadora de bolsillo conectada directamente a un multímetro electrónico, de forma que accedan los datos directamente a la misma.

El lenguaje de programación ha sido la gran dificultad en la aplicación de las calculadoras programables, dificultad que se ha solucionado de formas distintas, por lo cual es un hecho que cada fabricante tiene su particular manera de mandar la calculadora sin que exista una equivalencia fácil para un usuario que empieza a programar. Desde hace algu-

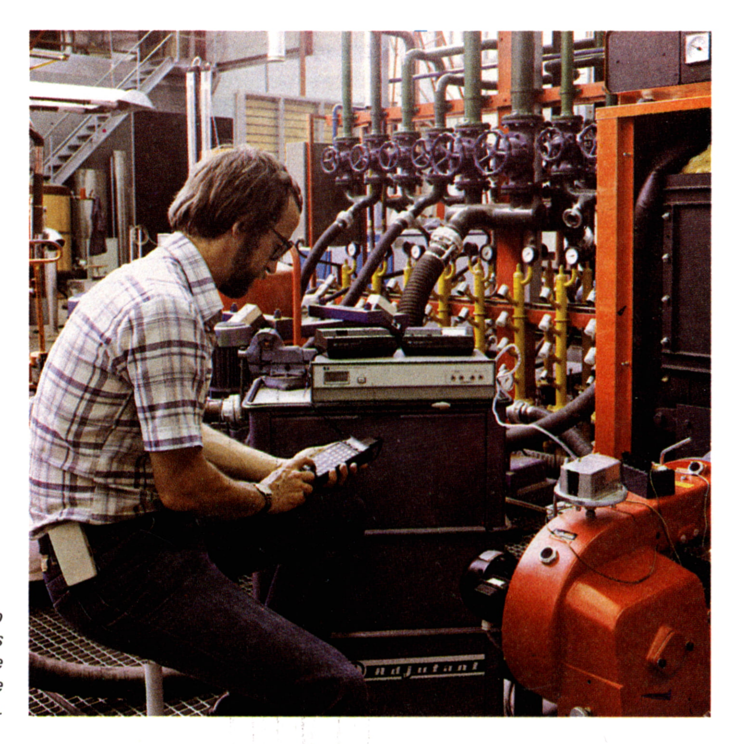
nos años se programan con los lenguajes de alto nivel como el BASIC, pero reduciendo el número de sus comandos y funciones. Quizás sea éste el lenguaje de las calculadoras programables de bolsillo, pero tiene que llegarse al precio adecuado para el bolsillo de los estudiantes, los cuales aprenden a programarlas ya en su formación de tipo medio: Bachillerato y Formación Profesional.

Una vez logrado el fácil transporte del computador de bolsillo se ha procurado interconectarlo directamente con aparatos de medida, con impresoras, lectores ópticos de



Toma simultánea de gran cantidad de muestras de la temperatura y humedad del suelo, gracias al programa de la calculadora de bolsillo que los procesa y almacena en una cassette digital.

códigos de barras, cassettes, etc. En la actualidad estamos asistiendo a un gran desarrollo de la conexión de la calculadora de bolsillo programable a otros aparatos electrónicos, de manera que la entrada de datos como su salida para el mando de relés o transistores de potencia sólo necesita un buen programa y que el operario pulse el botón de la puesta en marcha.



Toma de datos y mando de electroválvulas mediante el programa de una calculadora de bolsillo.

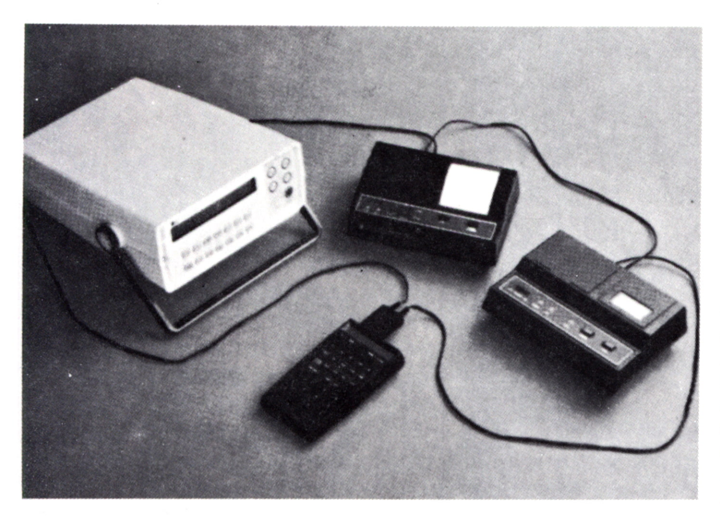
La fisonomía de la calculadora de bolsillo ha cambiado desde que empezó a competir con la regla de cálculo, pero era lógico que así fuera.

La gran tarea de la informática está en la entrada y salida de datos, ya que la electrónica de la pequeña calculadora hace el resto con el programa. El paso final suele consistir en dejar constancia del programa mediante una minúscula impresora incorporada a la calculadora.

TENDENCIAS EN LA CONSTRUCCION DE LAS CALCULADORAS DE BOLSILLO PROGRAMABLES

Desde que aparecieron las primeras calculadoras de bolsillo la miniaturización ha sido una tendencia general, pero ello ha chocado con las particularidades del teclado, que ha de tener un número de teclas determinado así como un tamaño en función de los dedos de la mano.

Primero se accionaron los comandos y funciones directamente con las teclas. Cada tecla tenía su función incorporada, por lo que al haber más de cien funciones que accionar se tuvo que asociar varias funciones a cada tecla con selección por otras que se distinguen por su color y también por teclas denominadas *inversas.* Así por la tecla de la función *seno* de un ángulo se obtiene el *arco seno* pulsando previamente la tecla INV.



Interconexión de una calculadora de bolsillo con un multímetro, una impresora y un cassette digital. El programa de la calculadora decide cuándo hay que tomar los datos, almacenarlos e imprimirlos.

Actualmente se prefiere asociar a las teclas las funciones que necesita el usuario, de manera voluntaria y con las funciones de un catálogo muy extenso que comprende los comandos y funciones de la unidad central y los de los periféricos que pueden conectarse a ella. Esta asociación a

gusto del usuario puede hacerse cómodamente por una tecla o por haberlas almacenado previamente en una tarjeta magnética. Con ello se llega a la *calculadora personal*, que tiene una carátula superponible sobre el teclado con las nuevas funciones asignadas.

El hecho de asociar funciones nuevas tiene la consecuencia de que la calculadora ha de ser alfanumérica, es decir, que usa tanto los números como las letras del alfabeto y con ello se pasa poco a poco a utilizar un lenguaje de alto nivel como el BASIC, ya sea asociando los comandos a las teclas o seleccionándolos por su escritura.

La disposición del teclado alfanumérico, como el de la



Calculadora financiera para estudiantes, hombres de negocios y analistas financieros. (Cortesía: Texas Instruments).

máquina de escribir, hace que la calculadora tome la forma alargada en horizontal, forma que permite alargar la longitud de la pantalla hasta 30 caracteres, lo cual es muy útil para las instrucciones de las líneas de BASIC.



La principal ventaja de las calculadoras programables está en que pueden trabajar en cualquier lugar y de forma autónoma, además de interconectarse con diversos periféricos. (Cortesía: Hewlett-Packard).

Sobre la construcción de las pantallas de salida de datos hay que indicar que al principio, por utilizarse LED, había un gran consumo de corriente, pero la adopción del cristal líquido que funciona con una corriente muy débil se ha hecho extensivo a todas las calculadoras, no solamente por el consumo eléctrico sino porque permite una gran variedad de caracteres, ya sean obtenidos con segmentos o por puntos en forma de matriz.

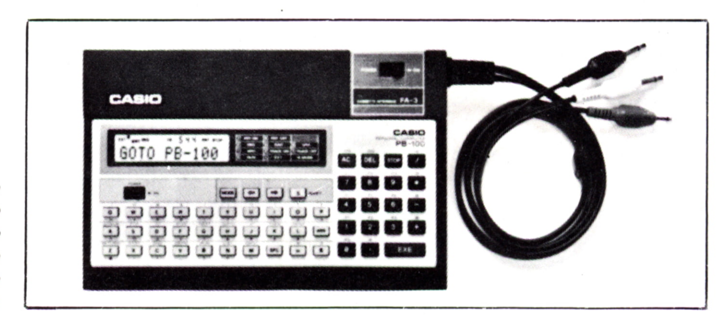
La memoria continua de datos y programa ha sido otra tendencia suministrada por los componentes MOS. Antiguamente, al desconectarla de la alimentación se perdían el programa y los datos obtenidos por el proceso de cálculo.

Por esta causa aparecieron calculadoras que tenían un lector de tarjetas magnéticas incorporado e inseparable de la unidad central. Con ello quedaban guardados los preciosos programas en pequeñas tarjetas magnéticas, pero más tarde la memoria continua y un aumento de su memoria RAM,

hizo que se pudiesen guardar de forma permanente, aunque el lector de tarjetas no desapareció sino que tomó la forma de periférico enchufable.

Los módulos preprogramados aparecieron ya hace tiempo y permitían disponer de una gran cantidad de programas especializados, principalmente de matemáticas, ya que la calculadora tiene las máximas facilidades para el tratamiento de los números.

No obstante, la posibilidad de conectarle estos módulos a través de una docena de contactos metálicos hizo que pronto se conectara la calculadora a una impresora. Con ello empiezan a aparecer los periféricos de entrada y salida de datos, generalmente acoplados por los mismos contactos que los módulos.



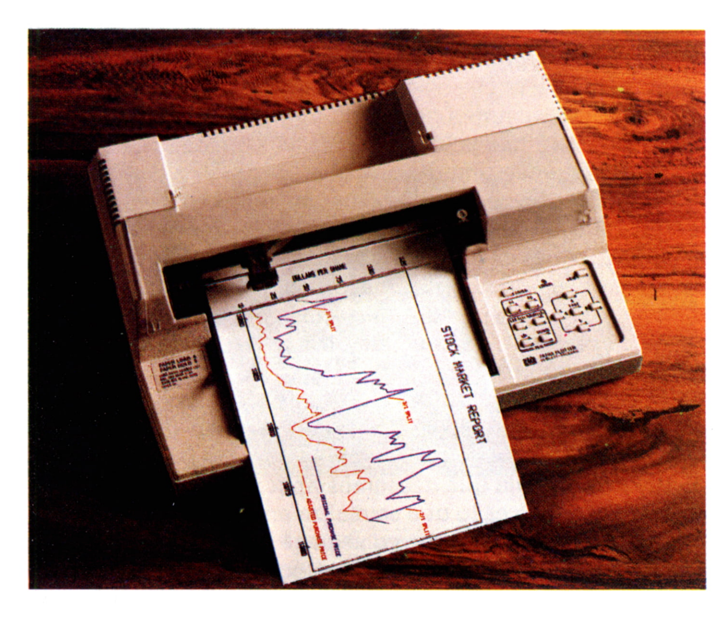
Para la conexión de una cassette de audio de cualquier marca hay que proporcionar a la calculadora la correspondiente interfase.

Además de la impresora, que está presente en todas las marcas por su utilidad en el listado de grandes programas y por la recogida de los datos procesados, como periféricos existen los que permiten entrar programas y datos en forma de códigos de barras, tienen la forma de un lápiz óptico que emite luz y la recibe después de reflejarse en la secuencia de barras delgadas y gruesas. Otros son los casettes de audio o digitales, que tienen más precisión, pero los más interesantes son los multímetros que permitén almacenar en la memoria de la calculadora los valores eléctricos más variados según lo ordena el programa, valores que luego pueden pasar a otro computador que no se podría desplazar adonde lo hace la calculadora de bolsillo.

Otros periféricos de entrada de datos son los relacionados con el tiempo, que permiten que la calculadora de bolsillo

tome nota del momento en que han ocurrido determinados hechos. Es posible que se sustiyan así los cronómetros de las oficinas de análisis de tiempos.

Además de las impresoras, las calculadoras se pueden conectar a las tablas trazadoras de gráficos o *plotters*, a unidades para el mando de relés o electroválvulas y a otros computadores más potentes a través de las correspondientes interfaces que aseguran un buen diálogo entre la calculadora y los periféricos.



Las trazadoras de gráficos (plotters) pueden gobernarse desde calculadoras programables de bolsillo.

Se nota en la actualidad otra tendencia: la de conectar a la calculadora con módulos programables de RAM pero con memoria continua debido a que tienen su propia alimentación eléctrica en forma de una minúscula pila.

Puesto que estos módulos tienen la posibilidad de quedar privatizados con alta seguridad se espera que aparezca una gran cantidad de programas en el comercio, ya que el programador precisa que se le abonen adecuadamente las horas empleadas en la programación. Esto solamente puede hacerse con tarjetas magnéticas que se desprivatizan con facilidad.

Por todo ello, vemos que la calculadora programable de bolsillo tiene su propia personalidad frente a los computadores personales por su tamaño, por su versatilidad, por su configuración personal y por promover la existencia de muchos programas.

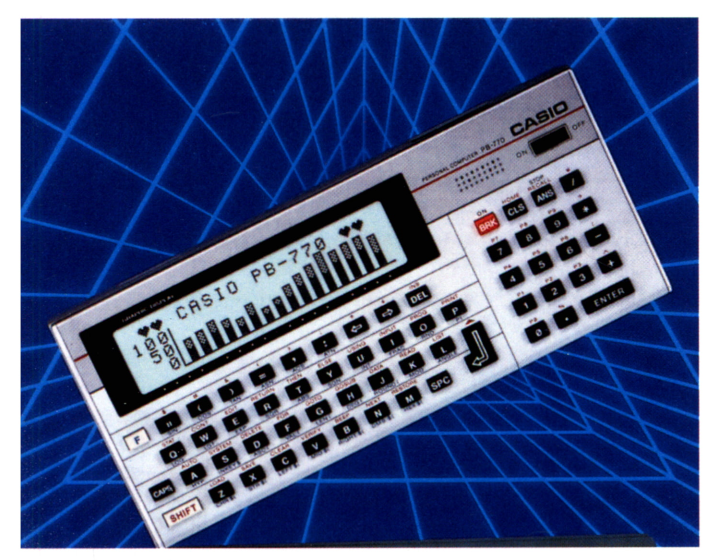
LENGUAJES UTILIZADOS EN LAS CALCULADORAS PROGRAMABLES

Como son muchas las funciones que puede hacer la calculadora programable, no es fácil organizar el programa que las encadena. Desde el principio se adoptó la forma que utilizaba la calculadora no programable. Así, si se tenían que sumar dos cantidades se pulsaba la tecla de sumar, pero aun con ello aparecen dificultades puesto que hay que hacer al menos tres cosas: entrar un sumando, entrar el otro y apretar la tecla de sumar. De esta forma ya puede quedar el resultado, pero por si hay más sumandos se prefiere seguir el orden de sumando, signo más, otro sumando, signo más, etc. que se termina con el signo igual para tener el resultado.

Ya se comprende que el instruir a la calculadora para que realice las operaciones que deseamos, dado que puede hacer más de cien funciones distintas, tiene que resultar un lenguaje algo complicado. Este lenguaje se denomina de *alto nivel* si se parece al natural, es decir, si ponemos en la pantalla la fórmula que deseamos aplicar y la calculadora se arregla sola. Esto es lo que se realiza en un lenguaje como el BASIC pero el tener que organizarse la calculadora por su cuenta, significa que ha de tener complicados programas internos que se pagan a precio de memoria RAM, por lo que a igualdad de capacidad de memoria las calculadoras en BASIC pueden procesar solamente programas unas 3 ó 4 veces más cortos.

Por otra parte, se denomina lenguaje de Informática de *bajo nivel* al que se acerca al único que puede entender la calculadora, el lenguaje binario. Como este lenguaje es sumamente incómodo, con ayuda de las teclas y sus funciones asociadas las casas constructoras logran un lenguaje también de bajo nivel que permite al usuario programarlas con más comodidad. Este lenguaje no tiene un nombre general porque cada constructor organiza la forma de programar sus calculadoras como le parece.

Por suerte son pocos los constructores de calculadoras programables y estos lenguajes se reducen a dos. Uno de ellos utiliza las ventajas de la denominada *Notación Polaca Inversa*, (RPN), mientras que los demás tienen un lenguaje parecido al algebraico que se denomina *Sistema Operativo Algebraico* (AOS).



Computador personal de bolsillo PB – 770 de la firma Casio. Incorpora 51 funciones científicas y puede trabajar con una gran cantidad de información gracias a la RAM que puede ampliarse hasta 32 kbytes. La pantalla admite 80 caracteres (20 columnas × 4 líneas, con gráficos de 160 × 32 puntos).

La Notación Polaca Inversa aparece por primera vez en una publicación del polaco Jan Lukasiewicz en 1929, bastantes años antes de que la Informática tuviese necesidad de utilizarla en los computadores, y consiste en operar algebraicamente sin emplear paréntesis a base de entrar dos operandos y luego la operación, guardando los resultados intermedios en unas memorias en forma de pila que se denominan STACK. Por otra parte, el lenguaje denominado Sistema Operativo Algebraico sigue las operaciones algebraicas tal como están escritas, empleando paréntesis si es necesario. Se comprende que este lenguaje sea más fácil de aprender, pero utiliza casi un 50 % más de memoria que el RPN y no hay que perder de vista que las calculadoras de bolsillo programables tienen poca memoria RAM utilizable por el usuario.

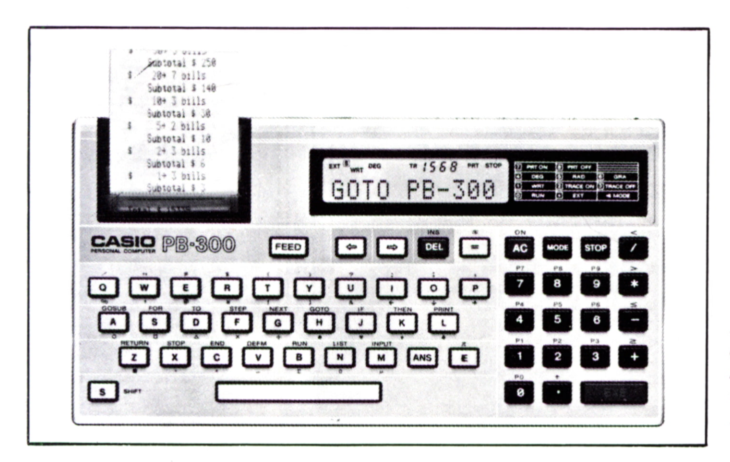
Veamos las características de estos lenguajes de bajo nivel. El Sistema Operativo Algebraico, dado que emplea las fórmulas tal como están, sólo tiene que cuidar de colocar los paréntesis cuando se presenta ambigüedad. Así, si tenemos que programar $A=B+C/D$, que puede ser $A=(B+C)/D$ o bien $A=B+(C/D)$, es necesario establecer una prioridad en las operaciones. Si la calculadora primero hace la operación de multiplicar o dividir y luego la de sumar o restar, sólo se necesitará el paréntesis cuando queramos realizar la operación $A=(B+C)/D$. Estas prioridades están indicadas en los manuales de las calculadoras con el nombre de *Jerarquía de las Operaciones.*



Terminal de video de edición, instalado en la redacción del periódico El País (Madrid y Barcelona).

La Notación Polaca Inversa es utilizada por el constructor de calculadoras Hewlett-Packard y necesita de las memorias intermedias STACK, que se denominan con las letras X, Y, Z, T, entrando el primer operando por la memoria X, al entrar el segundo, como estas memorias tienen la particularidad de que al entrar otro valor numérico el segundo queda en X mientras que, automáticamente, el primero pasa a Y como si fuese una pila de memorias que tienen entrada por la base, otras entradas harán ascender esta pila desde la base que es la memoria X y que es la que se visualiza en la pantalla. Al pulsar la tecla del operador se realiza la operación deseada

sin necesidad de accionar el signo igual, pues queda el resultado en la memoria X de la base, eliminándose el valor que estaba en la memoria Y y por lo tanto descendiendo los otros valores de Z y T. Estas calculadoras tienen la tecla denominada ENTER para separar dos cantidades que se entran juntas y para que con ello la primera ascienda a la memoria Y.



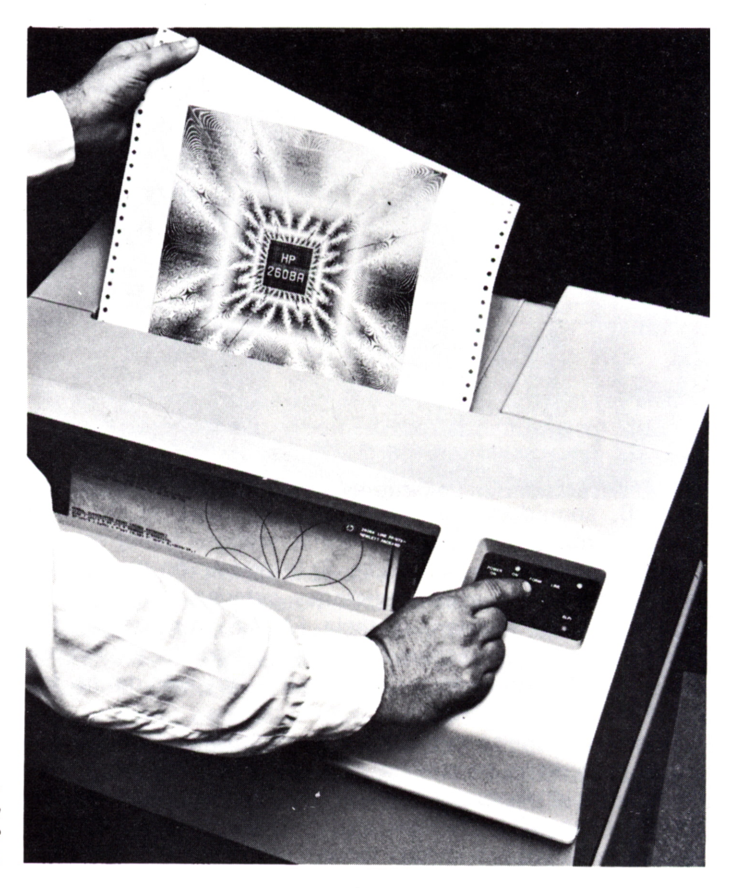
La impresora térmica o electrostática puede formar parte de la calculadora de bolsillo sin interconexión externa. (Cortesía: Casio).

Una ventaja de este sistema es que siempre tenemos en la pantalla los resultados intermedios y que además suprime los paréntesis y el signo igual. Así, si deseamos hacer la operación anterior A=(B+C)/D, pulsaríamos primero el valor B, seguidamente el de ENTER, luego el de C, a continuación el operador de la suma, luego el tercer valor D y finalmente la tecla de dividir. Si la operación se hubiese hecho por programa las funciones estarían en el mismo orden, pero se suprime la tecla ENTER porque los dos números ya quedan separados por el programa. Obsérvese que la función operadora viene siempre después de las dos cantidades o de sus resultados.

Posteriormente se verán ejemplos de programas en los tres lenguajes citados, pero ahora nos interesa comentar que los de bajo nivel necesitan en la programación detalles como dónde almacenar los resultados del programa, qué memorias se van a utilizar, con qué orden se van a reclamar los

resultados, etc. cuestiones que constituyen la técnica de la programación y de las que se liberan los lenguajes más cercanos al natural como el BASIC.

Los datos numéricos elaborados por la calculadora se almacenan en las memorias o registros, que en BASIC se denominan *variables*. Cada calculadora tiene un número limitado de ellas y generalmente se pueden intercambiar con los pasos que tiene el programa a razón de 8 pasos por cada memoria. Debido a esto, interesa utilizar el menor número de memorias a fin de disponer del máximo espacio para el programa.



Impresora Hewlett-Packard 2608 A utilizada como periférico de computador.

El cambio de memorias a pasos de programa se hace antes de escribir un programa con comandos propios de cada calculadora, como PARTICION o SIZE, que equivalen al DIM o DEFM de las calculadoras en BASIC. Para programar bien hay que consultar el manual que acompaña a la calculadora.



Tres modelos de calculadoras electrónicas de la firma Casio. La de la izquierda es de pantalla de cristal líquido con una sola fila de números, la del centro está preparada para resolver funciones científicas con capacidad para representar gráficos en pantalla, y la situada más a la derecha lleva incorporada una impresora.

Sobre las memorias es interesante destacar que con ellas se pueden realizar operaciones aritméticas. Así, si en el programa figura la instrucción STO + 2, indica que se sumará al valor que hay en la memoria 2 la cifra que precede a esta instrucción y que el valor de la suma quedará como nuevo valor de la memoria 2. Hay calculadoras que en vez de la citada instrucción emplean SUM 2, que si está precedida de la tecla INV será equivalente a STO − 2, y lo mismo para el producto PRD 2.

Otras particularidades sobre los lenguajes de bajo nivel se verán en el apartado siguiente, ya que aquí interesan solamente las diferencias en relación con su nivel. La mejor manera de entenderlo va a ser resolviendo ejemplos concretos.

COMO ESTABLECER EL PROGRAMA DE LAS CALCULADORAS

Dado que cada fabricante utiliza su lenguaje y a veces éste presenta diferencias en los distintos modelos de calculadoras, no es fácil resumir la programación para todas las calculadoras de bolsillo. Los ejemplos que se presentarán posteriormente serán tratados con calculadoras modernas, una para cada lenguaje, de las más sencillas y económicas a las más sofisticadas.



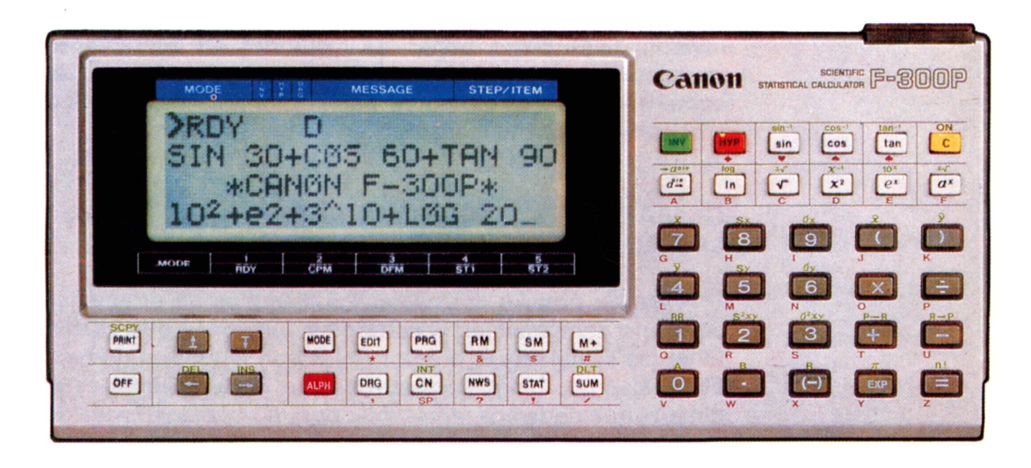
Representación gráfica estadística mediante un tubo de rayos catódicos y una impresora asociada a la computadora.

Para programar bien hay que proceder de forma ordenada, pues no hay que olvidar que a pesar de que un programa tenga cientos de comandos o funciones, si hay uno que esté equivocado no sirven para nada los demás. Para una correcta programación procederemos de la forma siguiente:

1) Una vez enterados del caso a programar lo reduciremos a unas reglas precisas, no ambiguas, que mediante un mínimo de pasos den un resultado único en un tiempo finito. Este paso de la programación se denomina poner la cuestión a programar en forma de *algoritmo*. Observése que no puede

quedar nada ambiguo ni obtener varios resultados distintos, como ocurre con el doble signo de la raíz cuadrada.

2) Con el algoritmo se construye en forma gráfica un *diagrama de flujo*, que nos permite comprender de un golpe de vista las interconexiones que tendrán las partes del programa. El diagrama de flujo suele tener tres símbolos: Un



círculo para comenzar y acabar, un rombo para las bifurcaciones condicionales y un rectángulo para las demás operaciones del proceso de datos, para la entrada y salida de datos y demás observaciones.

3) Se pasa este diagrama a las instrucciones del *listado* del programa, según el lenguaje de cada calculadora.

4) Se comprueba su funcionamiento con varios ejemplos de resultado conocido. Si el programa no funciona, se comprueban los resultados paso a paso y se mira si lo que aparece en la pantalla es lo que debe aparecer, hasta encontrar el paso de programa que opera mal.

5) Es importante anotar, cuando el programa ya funciona, los siguientes extremos en la hoja de programación: *A*) Las memorias utilizadas y su contenido. *B*) Las operaciones auxiliares que se hacen antes de ponerlo en marcha, tales como las condiciones iniciales, la participación de la memoria, si hay que dejar a cero estas memorias, etc. *C*) Hay que poner comentarios junto a las instrucciones del listado para recordar posteriormente su funcionamiento. *D*) Anotar

Calculadora científica Canon modelo F-300 P con pantalla de cristal líquido. Este modelo es capaz de resolver de forma gráfica problemas estadísticos.

también el código de las instrucciones o comandos de cada paso, puesto que es muy útil para comprobar si el programa se ha entrado bien en sucesivas ocasiones.

Respecto a las operaciones auxiliares del programa hay que hacer notar que si sobran pasos de programa es mejor poner todas las operaciones que se pueda en el listado, pero si la calculadora es de memoria pequeña hay que ejecutar varias operaciones antes de poner el programa en marcha, tales como poner las memorias en condiciones iniciales de su contenido, ir a un paso determinado al comenzar, etc. que como se hacen una sola vez, pueden efectuarse de forma manual.

COMANDOS Y FUNCIONES PARA LA PROGRAMACION

Antes de pasar a considerar los ejemplos de programas conviene comparar los comandos y funciones que suelen tener las calculadoras programables con lenguajes de bajo nivel. El único lenguaje que entiende la calculadora es el binario, que podemos representar por unos y ceros, forma de representar el paso de corriente eléctrica y la no conducción. Al accionar cada tecla se ponen en marcha programas internos que realizan la operación de la tecla, siempre en lenguaje binario. Por ello hay veces que la respuesta de la tecla no es instantánea, pero es que además este programa tiene que comprobar si se ha pulsado previamente la tecla de cambio de función, si se está en modo programa o de operación, si estamos en modo alfanumérico y si la función se pide escribiéndola por las teclas, si esta función está en el catálogo de funciones de la unidad central o de un periférico y, en este caso, si el periférico está acoplado a la unidad central. Tienen que producirse un montón de comprobaciones antes de que quede seleccionada la función o comando que deseamos.

La calculadora se pone en marcha por una tecla o interruptor en el que figuran las palabras ON/OFF. Si la dejamos conectada y sin utilizar durante unos minutos, se desconecta automáticamente para no consumir de forma exagerada las pilas o baterías.

Se pasa del modo manual al de programación accionando una tecla que a veces es PROGM, otras LRN, otras WRT.

Entonces se puede escribir y modificar un programa. El modo manual de trabajo se suele indicar con RUN.

En este modo manual podemos tener en la pantalla un número de cifras decimales que se eligen con FIX, aunque también existe la forma utilizada en ingeniería en grupos de tres decimales con ENG. Se borra el contenido de la pantalla con CLX, CE, DEL, y el de las memorias con CLRG, MAC, CLEAR, etc.

Comandos y funciones de las calculadoras programables

Sobre el modo de operar: ON, OFF, PRG, LRN, WRT, PRO, USER.
Sobre la forma de presentar la pantalla: FIX, ENG, SCI, AON, AOFF.
Sobre la forma de tratar las cifras: CHS, +/−, INT. FRC, ABS, RND.
Sobre unidades angulares y de tiempo: GRAD, RAD, DEG, P→R, R−D, HMS, DMS.
Sobre operaciones aritméticas y logarítmicas: $1/x$, X^2, $\sqrt{x}$, *SQRT,* y^x, *LOG, LN,* e^x, *MOD.*
Sobre operaciones estadísticas y trigonométricas: MEAN, $\bar{x}$, *SDEV, s, SIN, COS, TAN.*
Sobre el borrado de datos y programas: CLRG, CLEAR, MAC, CLX, CE, DEL, CLD, CLP.
Sobre el tratamiento de las memorias o registros de datos: STO, MIN, RCL, MR, STO+, SUM, STO×, PRD.
Sobre la programación: LBL, GTO, GOTO, GSB, GOSUB, RTN, RETURN, EXE, STOP, HLT, PAUSE, RUN, SF (flags), CF, FS?, DSE, ISG, DSZ.

Las cantidades de la pantalla pueden modificarse por cambio de signo con CHS o bien +/−, pero la parte entera queda con INT o INTG, mientras que la fraccionaria queda con FRC o FRAC. Lo mismo podríamos decir del valor absoluto que se obtiene con ABS.

Las funciones trigonométricas, así como las estadísticas,

suelen tener teclas perfectamente localizables como SIN, COS, TAN, $\bar{x}$, s, etc.

Para almacenar datos en las memorias se usa generalmente la palabra STO, pero determinadas calculadoras usan MIN, mientras que para reclamar los datos almacenados es normal utilizar RCL y MR. Ya se ha indicado que puede operarse aritméticamente con las memorias de la forma STO+ que para otras calculadoras es SUM, mientras que el



Utilizando pantallas TRC con alta resolución, es posible reproducir gráficos con gran precisión.

producto es PRD. El lenguaje RPN tiene funciones para las memorias de la pila o STACK del tipo STO Y, RCL T, STO+Z, que le dan una gran versatilidad.

En la programación puede modificarse el camino que sigue el programa con GTO o bien GOTO, y si se trata de subrutinas que regresan al paso siguiente de donde han salido, se utiliza GSB, GOSUB, EXE, con retorno mediante RTN, RETURN, o a veces INV 2nd SBR. Estos cambios del camino se realizan en forma condicional utilizando las condiciones del tipo x=y? o bien x=t, con todas las variantes de mayor, menor, distinto que, las cuales cada calculadora tiene de forma particular.



Modelo TI-66 de calculadora programable de bolsillo, provista de teclado alfanumérico de cristal líquido. (Cortesía: Texas Instruments).

Todo este conjunto de funciones y comandos debe poderse accionar con las teclas de la calculadora, que suele tener un máximo de 5 columnas por 9 filas para facilitar el código de cada tecla con un mínimo de cifras. Cuando existen muchas funciones y no es posible accionarlas con las teclas, se asocian a cada una pasando a modo USER, lo cual permite tener la calculadora a gusto del usuario.

ESTRUCTURAS BASICAS DE LAS ACCIONES A PROGRAMAR

Con todas estas funciones es posible programar los casos más complicados con la limitación de la memoria RAM que tenga cada calculadora. Se suele considerar que hay tres estructuras básicas de las acciones a programar. La *secuencial* muy utilizada para la programación de fórmulas, la *condicional* en la que hay una bifurcación en función de una condición y la *iterativa* que repite una acción un número de veces determinado.

Num.	Teclas	Comentarios (1)
00	STO 0	Entra cateto 1
01	R/S	Entra el 2
02	x^2	
03	+	Suma de cuadrados
04	RCL 0	
05	x^2	
06	=	
07	$\sqrt{x}$	Raiz de la suma
08	=	
09	R/S	Resultado en pantalla
10	RST	Ir al origen

Figura 19. Listado del programa n.º 1.

Hay calculadoras de bolsillo programables que por carecer de las funciones condicionales x=y, x>y, x=t, etc. solamente pueden programar acciones que sigan la estructura secuencial. Esta falta de posibilidades en la programación se nota porque no tienen las citadas funciones en su teclado ni la GTO, pudiéndose solamente conducir el programa al origen con el comando RST.

Aunque muchas veces se programan fórmulas algebraicas, el funcionamiento de las calculadoras de bolsillo tiene que comprender también la programación de estructuras condicionales y de las iterativas. Posteriormente se verán ejemplos de programación que se apartan de la sencilla programación de fórmulas, ya que al incluirse una condición

en el algoritmo y por ello en el programa, se obtienen los resultados adecuados.

Ejemplos de programas con la estructura secuencial

Los ejemplos de programas que figuran a continuación están tratados con calculadoras representativas de cada tipo de lenguaje. Para tener una mayor variedad en la aplicación de la programación a distintos casos se han elegido calculadoras desde la más sencilla como la Texas TI-57LCD, hasta la más completa como la Hewlett-Packard HP-41C, pasando por la más rápida en lenguaje BASIC, la Casio PB-100.



Pequeño computador científico de bolsillo fx=5200 P de Casio. Incluye 72 funciones cietíficas, que permiten cálculos de todo tipo. El lenguaje de programación es el Basic.

En los ejemplos de programas con estructura secuencial no se incluirán los diagramas de flujo por tratarse de una secuencia de acciones similar a la que figura en la parte derecha de los listados para la calculadora TI-57LCD. Así, para el programa n.º 1 de la figura 19 y para esta calculadora, supongamos que estamos en el caso de tener que hallar muchas veces el valor de la hipotenusa de un triángulo rectángulo. Para ello programamos su fórmula tal

como puede verse en el listado. Previamente a la utilización hay que entrar los valores de los catetos, el primero al empezar a utilizar el programa, luego se pulsa R/S para que por STO 0 quede almacenado. En la detención siguiente hay que entrar el segundo cateto, que ya se procesa sin entrarlo a ninguna memoria. El programa se detiene presentando el resultado de la hipotenusa y a continuación pueden entrarse nuevos valores de los catetos.

En la utilización, como solamente se usa la memoria 0, haremos la partición pulsando 2nd Part 1, con lo que quedarán 48 pasos de programa, luego pulsando LRN se entrará en modo de programa. Se borrará el programa anterior con 2nd CP y se entrarán los 11 pasos del programa (el primero tiene el número 00). Se volverá al modo de cálculo normal pulsando otra vez LRN, se irá al principio del programa con RST y se empezarán a entrar los valores de los catetos y pulsar R/S, tal como se ha indicado.

```
01◆LBL "HIP"
02 "CATETO 1="
03 PROMPT
04 STO 00
05 "CATETO 2="
06 PROMPT
07 X↑2
08 RCL 00
09 X↑2
10 +
11 SQRT
12 STOP
13 GTO "HIP"
14 .END.
```

Figura 21. Listado del programa n.º 2.

Esta secuencia de operaciones que hay que realizar antes de empezar a utilizar un programa tiene que hacerse siempre con la calculadora TI-57LCD, salvo el número de memorias que puede ser mayor. En programas posteriores esta secuencia de operaciones se indicará en forma abreviada.

Por el listado puede observarse que después de los cálculos hay que usar el signo igual para tener el resultado, y que además no hay ninguna bifurcación sino solamente RST que retorna al principio del programa.

El mismo caso de hallar la hipotenusa por el Teorema de

Pitágoras lo tenemos programado en RPN en el programa n.º 2 de la figura 21 para la calculadora HP-41C. Puede comprobarse que, por tratarse de una calculadora alfanumérica, los datos a entrar se anuncian en la pantalla por el comando PROMPT que detiene el programa y presenta el mensaje. Debido a que usa la Notación Polaca Inversa (RPN), después de entrados los valores y elevados al cuadrado se hace la suma en la operación 10 del listado, lo cual en el programa anterior se hacía en la operación 03 entre los sumandos.

```
10 PRINT "HIPOTENU        CATETO 1=?
   SA"                    3
20 INPUT "CATETO 1        CATETO 2=?
   =",C                   3
30 INPUT "CATETO 2        HIPOTENUSA= 4.242640
   =",D                   687
40 H=SQR (C*C+D*D)
50 PRINT "HIPOTENU
   SA=";H
60 GOTO 20
70 END
```

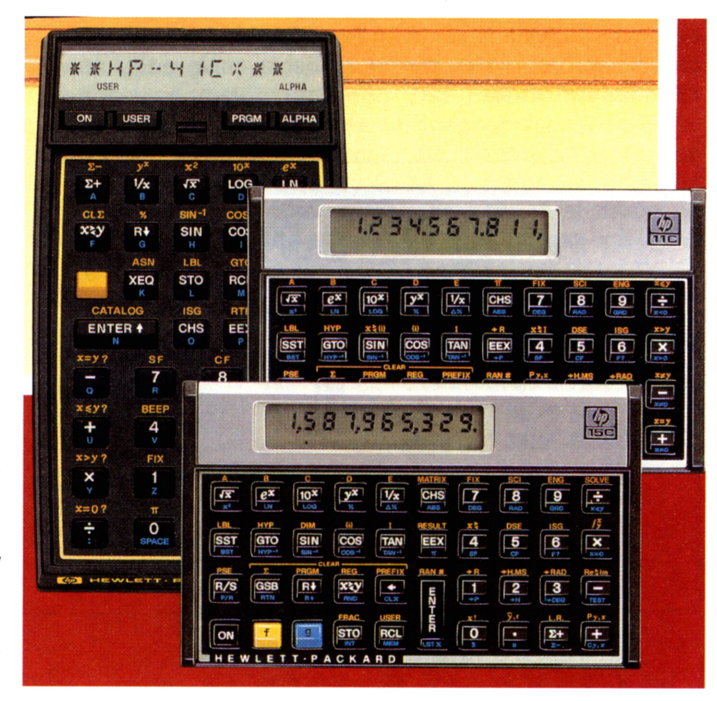
Figura 22. Listado del programa n.º 3.

Puede comprobarse que no existe el signo igual en este listado, porque la función suma se hace en la citada operación 10 una vez se han entrado los sumandos y han quedado almacenados en las memorias de la pila o STACK.

Esta calculadora va a etiquetas o LBL, empezando siempre con una etiqueta alfanumérica que se puede utilizar para regresar al origen, línea 13 del listado. Nótese como se indican los cuadrados en las líneas 07 y 09, así como la raíz cuadrada en la línea 11.

Un tercer programa para la obtención de la hipotenusa en la calculadora PB-100 lo tenemos en la figura 22, programa n.º 3, en lenguaje BASIC. Puede verse que también aquí hay rótulos que aparecen en la pantalla para

indicar los valores que hay que entrar. Estos valores pasan a las variables C y D, que hacen la función de las memorias de antes. Luego, en la línea 40 hay la fórmula del valor de la hipotenusa poniendo los cuadrados como productos, aunque también se podrían poner de otra forma.



Tres modelos de calculadoras profesionales de bolsillo. El modelo HP-11 C está diseñado para aplicaciones científicas y para técnicos e ingenieros, mientras que la HP-15 C lo está para cálculos matemáticos. El tipo HP-41 CX es un verdadero computador de bolsillo. (Cortesía: Hewlett Packard).

El resultado queda en la variable H y permanece en la pantalla porque el comando PRINT, además de dejar en pantalla este valor, detiene el programa. Puede observarse que por la impresión de un ejemplo, el resultado aparece en la misma línea que la palabra HIPOTENUSA por haberse puesto el punto y coma como separación. Para volver al origen se tiene que ir a un paso de programa, el 20 y con el comando GOTO. Queda por indicar que al principio de los programas hay un título que, como no tiene esta calculadora el comando REM, debe ponerse como PRINT con la consiguiente detención.

La utilización previa del programa para estas dos últimas calculadoras no tiene tantos pasos que recordar como la TI-57LCD. Para configurar el número de memorias a utilizar con la HP-41C se utiliza SIZE 002, mientras que con la PB-100, como normalmente tiene 26 variables que van de la letra A a la Z, no se indica ninguna configuración. Pero si se desea ampliarlas a un mayor número debe usarse el comando DEFM en grupos de 10 variables.

Respecto al borrado de los programas, la HP-41C lo efectúa con el comando CPL y la PB-100 con CLEAR. Ya se ha indicado que en la TI-57LCD el borrado de programa tiene que hacerse en modo LRN y con el comando 2nd CP. Hay que indicar finalmente que, como los programas empiezan asignando valores numéricos a las memorias y variables, no hace falta dejarlas a cero al inicio.

Num.	Teclas	Comentarios (4)
00	RCL 0	Dividendo
01	÷	
02	RCL 1	Divisor
03	=	
04	2nd Intg	Parte entera
05	R/S	Cifra cociente
06	x	
07	RCL 1	
08	=	
09	+/−	
10	+	Resta
11	RCL 0	
12	=	
13	x	
14	1	Aumenta un cero
15	0	
16	STO 0	
17	RST	Al origen

Figura 24. Listado del programa n.º 4.

Otro ejemplo de programación con estructura secuencial lo tenemos en la figura 24 programa n.º 4, que utiliza el algoritmo de la división para obtener el resultado del cociente, cifra tras cifra.

Por los comentarios de este programa se puede ver que la calculadora realiza los mismos pasos que realizamos al dividir.

Previamente, se coloca el dividendo en la memoria 0 y el divisor en la memoria 1. Luego el programa los divide, toma la parte entera como primer valor del cociente, halla el resto que con un cero a la derecha sirve para el nuevo dividendo y sigue repitiendo las mismas operaciones de nuevo. Aquí puede verse la ventaja de programar una secuencia de operaciones que se repetirán muchas veces y, como no hay bifurcaciones ni condiciones, estamos en la misma estructura que en el programa anterior, la secuencial.

```
811/241=
 3 3 6 5 1 4 5 2 2 8 2 1
 5 7 6 7 6 3 4 8 5 4 7 7
 1 7 8 4 2 3 2 3 6 5 1 4
 5 2 2 8 2 1 5 7 6 7 6 3
 4 8 5 4 7 7 1 7 8 4 2 3
 2 3 6 5 1 4 5 2 2 8 2 1


01♦LBL "DIV"
02 "DIVIDENDO="
03 PROMPT
04 STO 00
05 "DIVISOR="
06 PROMPT
07 STO 01
08♦LBL 05
09 RCL 00
10 RCL 01
11 /
12 INT
13 STOP
14 RCL 01
15 *
16 CHS
17 RCL 00
18 +
19 10
20 *
21 STO 00
22 GTO 05
23 END
```

Figura 25. Listado del programa n.º 5.

Para la calculadora en RPN, la HP-41C, puede verse en el programa n.º 5 de la figura 25 que la entrada de datos se hace por los mensajes de la pantalla, con lo que al repetir como antes los cálculos ya no se puede regresar al origen del programa. Para ello se coloca una etiqueta o LBL en la operación 08, y desde el final se retorna a este punto.

Puede observarse que no aparece el signo igual en el

listado, tal como se ha indicado en el programa anterior, y que el signo de multiplicar es un asterisco igual que en BASIC, operaciones 15 y 20. La función CHS es el cambio de signo para restar.

En ambas calculadoras se utilizan dos memorias, lo cual es necesario tener en cuenta para la partición de la memoria RAM en pasos de programa y número de memorias.

```
10 PRINT "DIVISION            DIVIDENDO=?
   "                          811
20 INPUT "DIVIDEND            DIVISOR=?
   O=",A                      241
30 INPUT "DIVISOR=             3,3 6 5 1 4 5 2 2 8
   ",B                         2 1 5 7 6 7 6 3 4 8
40 C=INT (A/B)                 5 4 7 7 1 7 8 4 2 3
50 PRINT C;                    2 3 6 5 1 4 5 2 2 8
60 A=(A-C*B)*10                2 1 5 7 6 7 6 3 4 8
70 GOTO 40                     5 4 7 7 1 7 8 4 2 3
80 END                         2 3 6 5 1 4 5 2 2 8
                               2 1 5 7 6 7 6 3 4 8
                               5 4 7 7 1 7 8 4 2 3
```

Figura 26. Listado del programa n.º 6:

El programa n.º 6 de la figura 26 para la calculadora PB-100 tiene también la entrada de datos por los mensajes de la pantalla. Las variables A y B equivalen a las memorias anteriores 00 y 01, pero interviene una variable auxiliar C que podría suprimirse a costa de complicar las fórmulas, principalmente la de la línea 60. En vez de retornar a una etiqueta como en el programa anterior se retorna a un paso de programa, el 40, ya que todo lo que le precede es la parte del programa de la entrada de datos.

En los tres programas anteriores se han obtenido los resultados en cifras de una en una, pero si en vez de multiplicar por 10 al final del programa lo hubiésemos multiplicado por 1.000, tendríamos el resultado en grupos de tres cifras. Puede aumentarse este número de cifras, pero si se pasa de la mitad de las que retiene en la pantalla la calculadora quedan resultados erróneos.

De la estructura secuencial se presenta un tercer caso. Deseamos comprobar cómo se acercan al valor del número *pi* los polígonos de número de lados crecientes e inscritos en una circunferencia, tomando su perímetro. Para ello partimos del cuadrado cuyo perímetro en función del radio unidad en $4\cdot\sqrt{2}$. Luego pasamos al polígono de doble número de lados, el octógono, cuyo lado calculamos por la fórmula $L=\sqrt{2-\sqrt{4-l^2}}$ en función del polígono anterior.

En el programa iremos procesando dos datos cada vez, datos que almacenaremos en dos memorias: en la memoria 0 estará el número de datos del polígono y en la memoria 1 el

Num.	Teclas	Comentarios (7)
00	4	
01	STO 0	Datos iniciales
02	2	
03	$\sqrt{x}$	
04	STO 1	
05	LBL 5	
06	2	Duplica los lados
07	STO × 0	
08	RCL 1	Formula lado mitad
09	x^2	
10	+/-	
11	+	
12	4	
13	=	
14	$\sqrt{x}$	
15	+/-	
16	+	
17	2	
18	=	
19	$\sqrt{x}$	
20	STO 1	
21	×	Valor perimetro
22	RCL 0	
23	÷	Valor PI
24	2	
25	=	
26	R/S	Resultado
27	GTO 5	

Figura 27. Listado del programa n.º 7.

valor del lado. El perímetro se obtendrá al multiplicarlas y el valor se acercará al número *pi* al dividir este producto por dos.

```
01♦LBL "INSCR"
02 4
03 STO 00
04 2
05 SQRT
06 STO 01
07♦LBL 05
08 2
09 ST* 00
10 RCL 01
11 X↑2
12 CHS
13 4
14 +
15 SQRT
16 CHS
17 2
18 +
19 SQRT
20 STO 01
21 RCL 00
22 *
23 2
24 /
25 VIEW X
26 GTO 05
27 END
```

```
3,061467457
3,121445153
3,136548483
3,140331213
3,141277519
3,141514825
3,141569585
3,141569585
3,141527863
3,141527864
3,140860234
3,140860234
3,151525325
3,108645431
2,930859020
4,144860575
0,000000000
0,000000000
```

Figura 28. Listado del programa n.º 8.

La programación de este caso para la calculadora TI-57LCD podemos verla en la figura 27 programa n.º 7. Los datos iniciales para las dos memorias citadas están en las primeras operaciones del listado. A partir de la etiqueta LBL

5 empieza el cálculo del valor del lado del polígono de doble número de lados, hasta que en la línea 21 se halla el perímetro y se divide por dos para obtener un valor que se va acercando al de *pi*, pero que luego se aparta por los errores que se obtienen en el cálculo de las raíces cuadradas.

En los dos programas siguientes para las calculadoras HP-41C y PB-100 se han impreso los resultados obtenidos, que podrían mejorarse si junto a cada valor se hubiese impreso el número de lados. El programa n.º 8 de la figura 28 muestra el listado y resultados de cálculo de la calculadora HP-41C en RPN, semejante al programa anterior. El resultado a imprimir se logra con el comando de la línea 25, que tiene la ventaja de dar el resultado en la pantalla y en la impresora, si está presente.

```
10 PRINT "POL.INSC
   R"
20 B=4:C=SQR 2
30 B=B*2
40 C=SQR (2-SQR (4
   -C*C))
50 D=B*C/2
60 PRINT D
70 GOTO 30
80 END
```

```
POL.INSCR
 3.061467459
 3.121445152
 3.136548491
 3.140331158
 3.141277258
 3.141513834
 3.141573027
 3.141587942
 3.141592949
 3.141594618
 3.141607969
 3.140860234
 3.141287533
 3.142996147
 3.149821337
 3.156631769
 3.210595196
 0
```

Figura 29. Listado del programa n.º 9.

El programa n.º 9 de la figura 29 sigue los mismos pasos que los anteriores, pero los resultados presentan alguna diferencia por la forma que tiene cada calculadora de realizar los cálculos.

Estos ejemplos de programas con estructura secuencial muestran la sencillez de la programación en los tres

lenguajes con que podemos dialogar con las calculadoras programables de bolsillo.

Ejemplos de programas con la estructura condicional

Una vez que se programan las otras dos estructuras, generalmente se utiliza también la secuencial para sacar el máximo partido de la programación y entonces tiene que

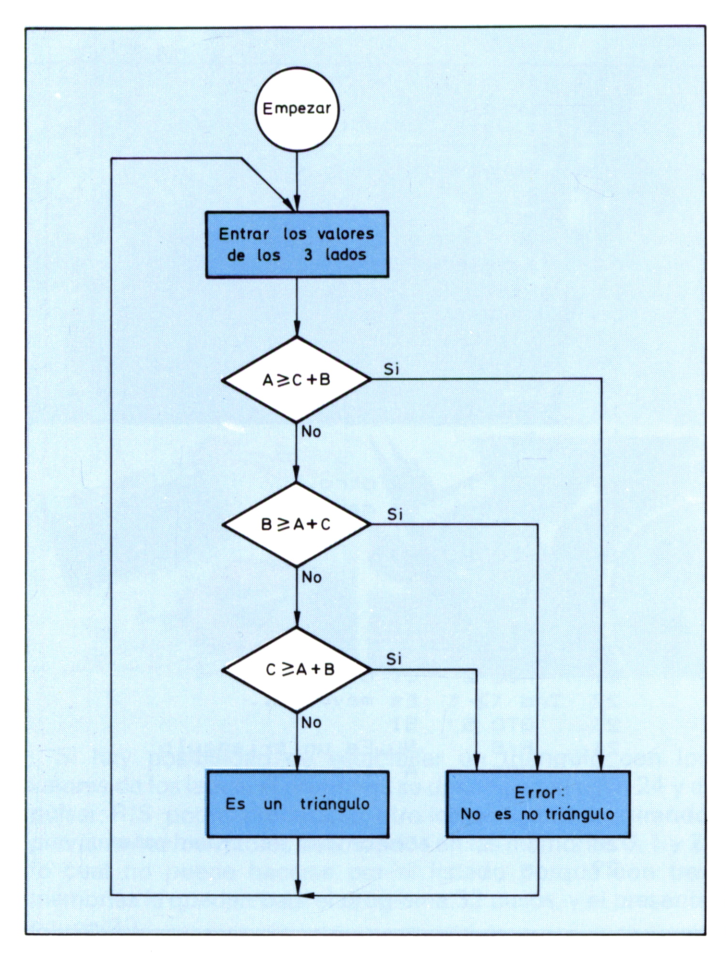


Figura 30. Diagrama de flujo de los programas números 10, 11 y 12.

utilizarse el diagrama de flujo para comprender con facilidad las interconexiones de las partes del programa.

El primer caso a programar con la estructura condicional se refiere a la comprobación por la calculadora de si con tres lados dados se puede obtener un triángulo. Para ello se realiza por tres veces la comprobación de si un lado es mayor que la suma de los otros dos.

Podemos ver este algoritmo interpretado en el diagrama de flujo de la figura 30.

Para el programa n.º 10 de la figura 31 los comentarios

Num.	Teclas	Comentarios (10)
00	RCL 1	
01	+	
02	RCL 2	
03	=	Suma de dos lados
04	X⇄t	
05	RCL 0	El otro lado
06	2nd X≥t	Es mayor que la suma?
07	GTO 5	SI
08	RCL 0	NO
09	+	
10	RCL 2	
11	=	
12	X⇄t	
13	RCL 1	El otro...
14	2nd X≥t	Es mayor...
15	GTO 5	SI
16	RCL 1	NO
17	+	
18	RCL 0	
19	=	
20	X⇄t	
21	RCL 2	El otro...
22	2nd X≥t	Es mayor ...
23	GTO 5	SI
24	R/S	NO.Es un triangulo
25	RST	A otro caso
26	LBL 5	
27	÷	
28	0	Error.No es triangulo
29	=	

Figura 31. Listado del programa n.º 10.

del listado permiten ver su equivalencia con el diagrama de flujo. Hay que indicar que para la calculadora TI-57LCD antes de establecer la condición de las líneas 06, 14, 22, hay que almacenar el valor correspondiente en la memoria *t*, lo cual se hace en las operaciones 04, 12 y 20. Si hay error debido a que no es un triángulo, se manda al LBL 5 que presenta un mensaje de error por el hecho de dividir cualquier cantidad por cero.







Las aplicaciones de las calculadoras programables de bolsillo se encuentran tanto en el campo científico como en el industrial y técnico. (Cortesía: Hewlett Packard).

Si hay posibilidad de establecer un triángulo con los valores de los lados, el programa se detiene en el paso 24 y al pulsar R/S podrá procesarse otro caso, siempre entrando previamente los valores de los lados en las memorias 0, 1 y 2, lo cual no puede hacerse por el listado porque con tres memorias le quedan para el programa 32 pasos, y el presente ocupa 30.

```
01♦LBL "TRIANG"
02♦LBL 04
03 "TRES LADOS="
04 PROMPT
05 STO 00
06 STOP
07 STO 01
08 STOP
09 STO 02
10 RCL 00
11 RCL 01
12 RCL 02
13 +
14 X<=Y?
15 GTO 05
16 RCL 01
17 RCL 00
18 RCL 02
19 +
20 X<=Y?
21 GTO 05
22 RCL 02
23 RCL 01
24 RCL 00
25 +
26 X<=Y?
27 GTO 05
28 "TRIANGULO"
29 PROMPT
30 GTO 04
31♦LBL 05
32 0
33 /
34 END
```

Figura 33. Listado del programa n.º 11.

Para la calculadora HP-41C puede utilizarse el programa n.º 11 de la figura 33, que facilita la entrada de datos por los mensajes y paradas en las líneas 03, 06 y 08. El programa sigue como el anterior el diagrama de flujo citado, pero

```
 10 PRINT "TRIANG."
 20 INPUT A
 30 INPUT B
 40 INPUT C
 50 IF A≥C+B THEN 9
    0
 60 IF B≥C+A THEN 9
    0
 70 IF C≥A+B THEN 9
    0
 80 PRINT "TRIANGUL
    O":GOTO 20
 90 PRINT "ERROR":G
    OTO 20
100 END
```

Figura 34. Listado del programa n.º 12.

como interesa tener la condición X< = Y? presente en el teclado de esta calculadora y no la que tenía la anterior, hay que cambiar los términos de la comparación.

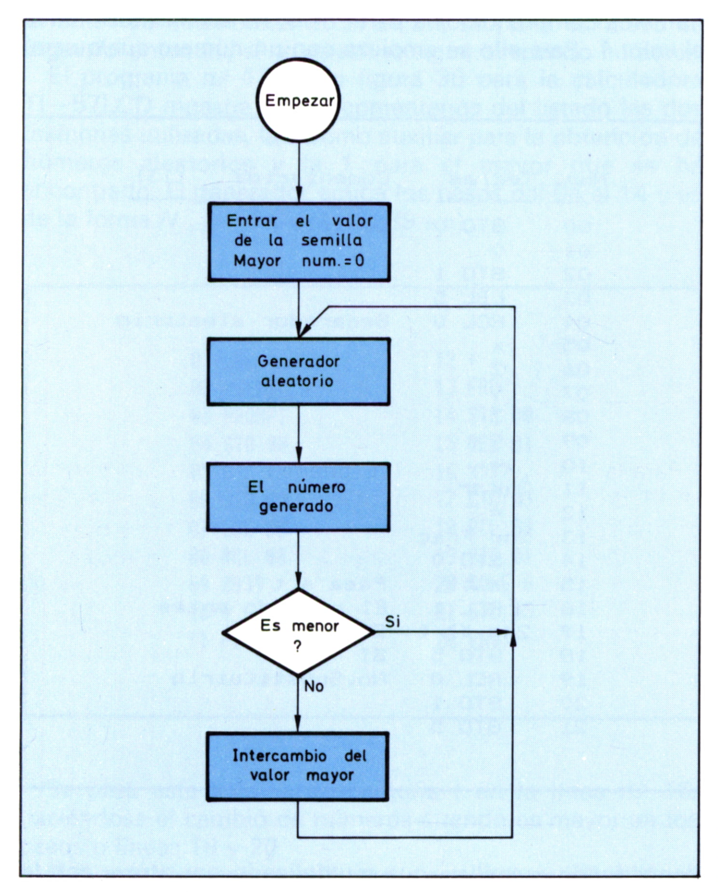


Figura 35. Diagrama de flujo de los programas números 13, 14 y 15.

Puede verse en el programa n.º 12 de la figura 34 para la calculadora PB-100 la facilidad con que se transforma el diagrama de flujo a los pasos del programa, lo cual suele ser general en la programación de casos algebraicos, pero no en otros en los que no está tan preparado el lenguaje BASIC.

Otro ejemplo de programa que utiliza la estructura condicional lo tenemos en el diagrama de flujo de la figura 35, por el que deseamos comprobar cuál es el mayor número aleatorio obtenido por un generador determinado.

Los generadores aleatorios proporcionan normalmente números comprendidos a partir de 0, no alcanzando nunca el valor 1. Para ello se empieza con un número cualquiera,

Num.	Teclas	Comentarios (13)
00	STO 0	Semilla
01	0	
02	STO 1	
03	LBL 5	
04	RCL 0	Generador aleatorio
05	x	
06	2	
07	9	
08	3	
09	9	
10	+	
11	2nd π	
12	=	
13	2nd Frac	
14	STO 0	
15	X⇄t	Pasa a t
16	RCL 1	El hallado antes
17	2nd X≥t	Es mayor?
18	GTO 5	SI
19	RCL 0	No.Substituirlo
20	STO 1	
21	GTO 5	

Figura 36. Listado del programa n.º 13.

denominado «semilla», que multiplicado por otro y con la suma de un tercero que tiene muchos decimales queda un resultado del que se utiliza solamente la parte fraccionaria y que es el número aleatorio generado. Este número ocupa el lugar de la anterior «semilla» y así se van generando números que no son del todo al azar, por lo que estos generadores se llaman *pseudoaleatorios.*

Con el mayor número aleatorio generado podemos

comprobar al cabo de mucho tiempo de generar números si se acerca al formado por nueves después de la coma decimal, o si se estabiliza por seguir un ciclo cerrado.

Por el diagrama de flujo podemos ver que después de obtenido un número aleatorio se comprueba si es mayor que el obtenido anteriormente. Si es así, se pone el nuevo en lugar del anterior y si no es así, se sigue buscando números.

El programa n.º 13 de la figura 36 para la calculadora TI−57LCD muestra en los comentarios del listado las dos memorias utilizadas, la 0 como auxiliar para la obtención de números aleatorios y la 1 para el mayor que se ha encontrado. El generador ocupa los pasos del 04 al 14 y es de la forma $N_{i+1} = \text{FRC}\ (N_i \times 2939 + \pi)$.

```
01♦LBL "ALEA"        12 +
02 "SEMILLA="        13 FRC
03 PROMPT            14 STO 00
04 STO 00            15 RCL 01
05 0                 16 X>Y?
06 STO 01            17 GTO 05
07♦LBL 05            18 RCL 00
08 RCL 00            19 STO 01
09 2939              20 TONE 6
10 *                 21 GTO 05
11 PI                22 END
```

Figura 37. Listado del programa n.º 14.

Se pasa este valor a la memoria *t* en la línea n.º 15, haciéndose el cambio de números cuando es mayor en los pasos o líneas 19 y 20.

El mismo programa queda para la calculadora HP-41C, tal como puede verse en el programa n.º 14 de la figura 37. Tiene el paso 20 una señal sonora que nos avisa de que se ha encontrado un número aleatorio mayor que los obtenidos hasta el momento, puesto que a medida que van apareciendo números mayores se tarda más tiempo en obtener otro.

Para la calculadora PB-100 el programa equivalente está en la figura 38 y puede servir, lo mismo que los anteriores,

```
10 PRINT "MAYOR AL
   EAT"
20 INPUT "SEMILLA=
   ",N:M=0
30 N=FRAC (N*2939+
   π)
40 IF M>N THEN 30
50 M=N:GOTO 30
60 END
```

Figura 38. Listado del programa n.º 15.

para ensayar nuevos generadores aleatorios cambiando los valores, introduciendo otras funciones, etc. lo cual se facilita en este lenguaje puesto que el generador está en la línea 30

```
01♦LBL "PASQ"
02 FIX 0
03♦LBL 01
04 "ANNO?"
05 PROMPT
06 STO 00
07 CLA
08 1900
09 ST- 00
10 RCL 00
11 19
12 /
13 FRC
14 19
15 *
16 RND
17 STO 01
18 7
19 *
20 1
21 +
22 19
23 /
24 INT
25 STO 02
26 RCL 01
27 11
28 *
29 4
30 +
31 RCL 02
32 -
33 29
34 /
35 FRC
36 29
37 *
38 RND
39 STO 03
40 RCL 00
41 4
42 /
43 INT
44 STO 04
45 RCL 00
46 +
47 31
48 +
49 RCL 03
50 -
51 7
52 /
53 FRC
54 7
55 *
56 CHS
57 RCL 03
58 -
59 25
60 +
61 X>0?
62 GTO 00
63 31
64 +
65 ARCL X
66 "⊦ MARZO"
67 PROMPT
68 GTO 01
69♦LBL 00
70 ARCL X
71 "⊦ ABRIL"
72 PROMPT
73 GTO 01
74 END
```

Figura 39. Listado del programa n.º 16.

del listado. Queda por indicar que los números aleatorios tienen muchas aplicaciones en distintas ramas de la ciencia y de la técnica, pero que al ensayar generadores hay que hacerlo con distintas «semillas» puesto que para algunas aparecen recurrencias que proporcionan pocos números al azar.



Módulo de arranque automático y duplicación de cassettes del HP-41. Este modelo permite la ejecución de un programa en cuanto se conecta el ordenador.
(Cortesía: Hewlett Packard).

Un tercer programa, algo más largo por lo que no es posible programarlo con la calculadora TI-57LCD, nos da el día de la Pascua de Resurrección de cualquier año entre 1900 y 2099.

Es un ejemplo de combinación de la estructura secuencial y de la condicional, puesto que después de distintos pasos del algoritmo debido a Gauss, se obtiene un resultado positivo si se refiere al mes de abril y negativo si es de marzo. Puede ensayarse su funcionamiento teniendo en cuenta que la Pascua más tardía corresponde al 25 de abril de los años 1943 y 2038, pero la más temprana cae fuera de las posibilidades del programa porque corresponde al 22 de marzo de los años 1818 y 2285.

Para la calculadora HP-41C puede verse en el programa n.º 16 de la figura 39 que después de programados los

pasos del algoritmo de Gauss, en la línea 61 se bifurca condicionalmente dando el mes y día de la fecha según sea positivo o negativo el resultado.

Como utiliza 5 memorias la partición inicial tiene que ser de SIZE 006. En las líneas 16 y 38 hay la función RND que elimina por redondeo las cifras finales de la parte decimal del valor obtenido.

```
10 PRINT "PASCUA"                PASCUA
20 INPUT "ANNO=",A               ANNO=?
30 D=A-1900:B=FRAC (             1943
D/19)*19                          25 ABRIL
40 C=INT ((7*B+1)/19             ANNO=?
)                                2038
50 R=FRAC ((11*B-C+4              25 ABRIL
)/29)*29                         ANNO=?
60 Q=INT (D/4)                   1964
70 E=FRAC ((D+Q-R+31              29 MARZO
)/7)*7                           ANNO=?
80 F=25-R-E:IF F>0 T             1983
HEN 100                           3 ABRIL
90 F=F+31:PRINT F;"              ANNO=?
MARZO":GOTO 20                   1984
100 PRINT F;" ABRIL"              22 ABRIL
:GOTO 20                         ANNO=?
110 END
```

Figura 41. Listado del programa n.º 17.

El mismo algoritmo está programado en lenguaje BASIC para la calculadora PB-100 en el programa n.º 17 de la figura 41. Por los ejemplos que se acompañan puede verse como obtiene el resultado en su día y mes por la bifurcación condicional de la línea 80.

Con los ejemplos precedentes se ha podido ver cómo aumentan las posibilidades de la pequeña calculadora programable de bolsillo al utilizar la estructura condicional junto con la secuencial.

Ejemplos de programas con la estructura iterativa

Como en la programación con la estructura iterativa repetimos una acción las veces necesarias hasta que se cumple una condición, es frecuente encontrar en estos programas una combinación de las demás estructuras, de tal manera que aparecen procesos secuenciales junto con los condicionales para realizar varias veces el proceso iterativo. Debido a ello, en los distintos lenguajes se facilita su



Figura 42. Las calculadoras programables de bolsillo permiten resolver de forma rápida cálculos repetitivos, basados en fórmulas de aplicación directa.

utilización con comandos específicos como el FOR... NEXT del lenguaje BASIC, el DSZ de la calculadora TI-57LCD y el DSE e ISG de la HP-41C, pero no es imprescindible su utilización ya que puede sustituirse por bifurcaciones condicionales que tienen el inconveniente de alargar la programación.

La figura 42 es un claro ejemplo de utilización de las calculadoras programables de bolsillo.

Esto podemos verlo en el primer ejemplo a programar. Al obtener el factorial de un número, el número de cifras crece tan rápidamente que pronto ya no cabe en la pantalla de la calculadora y entonces el resultado pasa a expresarse en la

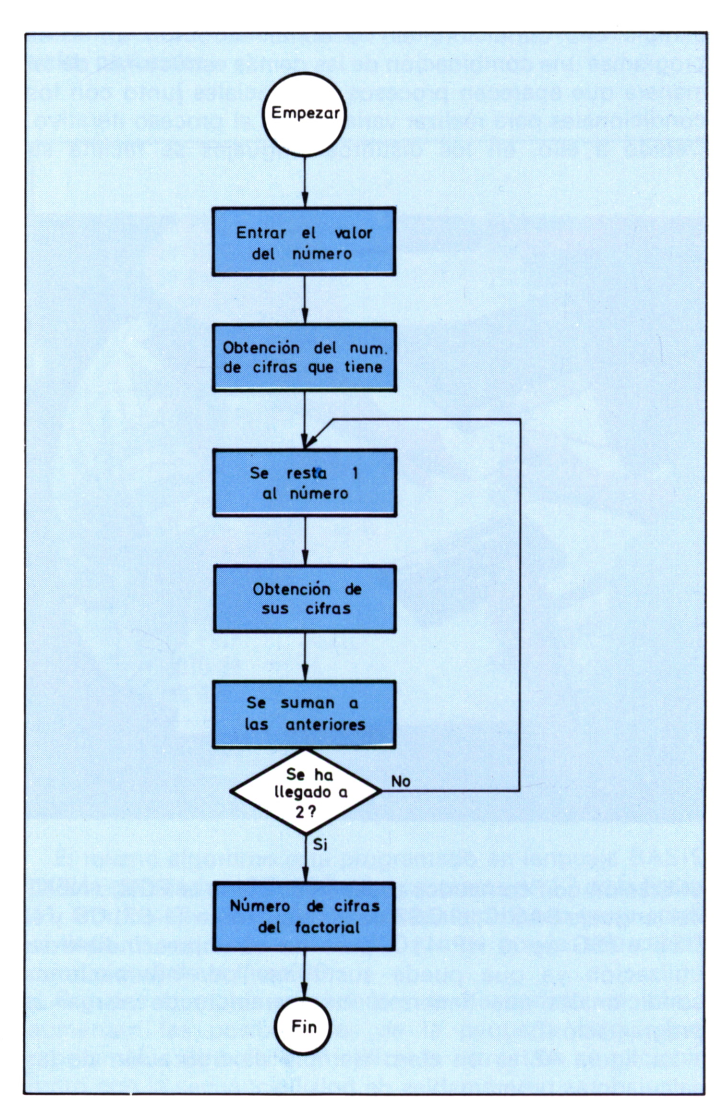


Figura 43. Diagrama de flujo de los programas 18, 19 y 20.

forma denominada de coma flotante. Pero con un exponente de diez, de 99 solamente se llega a obtener el factorial de 69. Hay ocasiones en que interesa saber el número de cifras que tiene el resultado de un factorial de un número y para ello establecemos el programa representado en el diagrama de flujo de la figura 43, que utilizando la función logaritmo va obteniendo el número de cifras que tiene.

```
Num.  Teclas      Comentarios   (18)
----  ---------   -----------------
 00      STO 0    Entra el numero
 01      2
 02      X⇄t      Referencia
 03      RCL 0
 04      Log
 05      STO 1    Primer valor
 06      LBL 5    Bucle cifras
 07      1
 08    STO - 0    Resta 1
 09      RCL 0
 10      Log      Cifras
 11    STO + 1
 12      RCL 0
 13   2nd X ≥ t   Es el final?
 14      GTO 5    No
 15      RCL 1    Si
 16   2nd Intg
 17      +
 18      1
 19      =
 20      R/S      Resultado final
```

Figura 44. Listado del programa n.º 18.

En el programa para la calculadora TI-57LCD, número 18, de la figura 44 podemos ver en los comentarios del listado que se toma la referencia del proceso iterativo en las líneas 01 y 02 del listado, y que se repite la suma de cifras obtenidas con la función LOG en la línea 11.

Una vez se ha llegado al final del proceso el número de cifras se obtiene tomando la parte entera de la suma y añadiéndole la unidad, resultado que queda en la pantalla al detenerse el programa.

```
01♦LBL "FACT"
02 FIX 0
03 "NUM="
04 PROMPT
05 STO 00
06 LOG
07 STO 01
08♦LBL 05
09 1
10 ST- 00
11 RCL 00
12 LOG
13 ST+ 01
14 RCL 00
15 2
16 X<=Y?
17 GTO 05
18 RCL 01
19 INT
20 1
21 +
22 END
```

Figura 45. Listado del programa n.º 19.

Para la calculadora HP-41C el programa equivalente es el número 19 de la figura 45, que sigue los mismos pasos que el anterior excepto en la colocación de la referencia que está dentro del bucle iterativo, en la línea 15, debido a que lo permite el lenguaje de la calculadora por establecer las comparaciones entre las memorias X e Y del conjunto de memorias de la pila o STACK.

```
10 PRINT "CIF.FACT
   ."
20 INPUT "NUM=",N:
   B=0
30 FOR A=N TO 1 ST
   EP -1
40 B=B+LOG A
50 NEXT A
60 C=INT B+1
70 PRINT "CIFRAS="
   ;C
80 GOTO 20
90 END
```

```
CIF.FACT.
NUM=?
50
CIFRAS= 65
NUM=?
100
CIFRAS= 158
NUM=?
150
CIFRAS= 263
NUM=?
200
CIFRAS= 375
```

Figura 46. Listado del programa n.º 20.

Al pasarlo al lenguaje BASIC para la calculadora PB-100, programa n.º 20 de la figura 46, se ha usado el bucle FOR...NEXT entre los pasos 30 y 50. Figuran junto al listado ejemplos en los que se puede ver cómo crece el número de cifras de un factorial.

Num.	Teclas	Comentarios (21)
00	STO 1	Dividendo
01	R/S	
02	STO 2	Divisor
03	2nd C.t	t=0
04	LBL 5	
05	RCL 1	
06	÷	Division
07	RCL 2	
08	=	
09	2nd Intg	
10	×	
11	RCL 2	
12	=	
13	+/−	Resta
14	+	
15	RCL 1	
16	=	
17	STO 0	
18	2nd X=t	El resto es cero?
19	GTO 6	SI
20	RCL 2	NO
21	STO 1	
22	RCL 0	
23	STO 2	
24	GTO 5	
25	LBL 6	
26	RCL 2	Resultado
27	R/S	
28	RST	Otro caso

Figura 47. Listado del programa n.º 21.

El siguiente ejemplo a programar es el clásico Algoritmo de Euclides para la obtención del máximo común divisor entre dos números. El proceso consiste en dividirlos y comprobar si el resto es cero. Si lo es, el resultado es el

divisor y si no, se cambia el divisor que se pasa al dividendo y el resto pasa al divisor, y se vuelve a dividir.

No se incluye el diagrama de flujo para este algoritmo, que se puede seguir fácilmente por los comentarios del listado del programa n.º 21 de la figura 47, para la calculadora TI-57LCD. La referencia para el final de la iteración, el resto nulo, se toma en la línea 03 y el cambio de memorias para cuando el resto no es nulo está en los pasos del 20 al 23. El resultado queda en la pantalla por el LBL 6 al detenerse el programa en la línea 27.

```
01♦LBL "MCD"
02 "DIVIDENDO="
03 PROMPT
04 STO 01
05 "DIVISOR="
06 PROMPT
07 STO 02
08♦LBL 05
09 RCL 01
10 RCL 02
11 /
12 INT
13 RCL 02
14 *
15 CHS
16 RCL 01
17 +
18 STO 00
19 X=0?
20 GTO 06
21 RCL 02
22 STO 01
23 RCL 00
24 STO 02
25 GTO 05
26♦LBL 06
27 RCL 02
28 END
```

Figura 48. Listado del programa n.º 22.

Para la calculadora HP-41C queda este algoritmo programado en la figura 48 programa n.º 22. Los valores se entran según los mensajes que presenta la pantalla y se sigue la misma programación que en el caso anterior, utilizando también tres memorias, dato a tener en cuenta para la partición inicial de la memoria.

Para la calculadora PB-100 el mismo programa se puede ver en la figura 49, con el número 23, y puede comprobarse que la condición final de la iteración está en la línea 50. Este programa se va a utilizar como subrutina (o programa auxiliar) del que sigue a continuación.

```
10 PRINT "M.C.D."
20 INPUT B
30 INPUT C
40 E=INT (B/C):R=B
   -C*E
50 IF R=0 THEN 70
60 B=C:C=R:GOTO 40
70 PRINT "M.C.D.="
   ;C
80 GOTO 20
90 END
```

Figura 49. Listado del programa n.º 23.

El tercer caso de programación con estructura iterativa tiene una mayor complejidad, pero con una ligera modificación cabe en los pocos pasos de la calculadora TI-57LCD. Se trata de obtener tres números enteros que pueden ser lados de un triángulo rectángulo y que por el teorema de Pitágoras se denominan *ternas pitagóricas.* Un ejemplo de ellos lo tenemos con los números 3, 4 y 5, ya que si se elevan al cuadrado la suma de los dos primeros da el tercero.



Interconexión de una calculadora con una impresora y un cassette de audio.

Hay 18 *ternas pitagóricas* utilizando números hasta el 100, pero también aparecen los múltiplos y según como se programe, lo hacen en orden inverso, es decir: el 4, 3 y 5.

```
Num. Teclas     Comentarios    (24)
---- ---------  ------------------
 00    RCL 1    Primer numero
 01    -
 02    1        Menos uno
 03    =
 04    STO 0    Pasa al segundo
 05    RCL 1
 06    x²
 07    STO 2    Auxiliar
 08    LBL 1
 09    RCL 0
 10    x²
 11    +
 12    RCL 2
 13    =
 14    √x
 15  2nd Frac
 16    X⇄t
 17    0
 18  2nd X=t    Cuadrado perfecto?
 19  2nd SBR 2. SI
 20  2nd Dsz    NO.Disminuye 1
 21    GTO 1
 22    1
 23    STO -1   Disminuye 1
 24    RST      Al origen
 25    LBL 2
 26    RCL 1    Resultados
 27    R/S
 28    RCL 0
 29    R/S
 30 2nd INV SBR
```

Figura 51. Listado del programa n.º 24.

Para evitar estas repeticiones el programa analiza cada uno de los cien primeros números, pero al comprobar si con el segundo se establece una *terna pitagórica*, se empieza con el número siguiente. Y para evitar que no aparezcan los múltiplos, una vez obtenida una *terna* se aplica la condición

del *algoritmo de Euclides.* Esta condición es la que no cabe en el programa de la calculadora TI-57LCD, puesto que al utilizar tres memorias quedan para el programa 32 pasos.

Ayudan a la comprensión del funcionamiento del programa, n.º 24 de la figura 51, los comentarios del listado y por ellos vemos que esta calculadora solamente tiene el comando DSZ para la memoria O, que se utiliza para la repetición de pruebas con el segundo número de las *ternas* en la línea 20. Para utilizar este programa se empieza con una partición de la memoria de tres y 32 pasos de programa. Luego se inicia poniendo 100 STO 1, para que vayan procesándose todos los números menores de 100 hasta llegar a cero, pero como se trata de una calculadora lenta tarda bastante tiempo en obtener las *ternas pitagóricas.*



El cassette digital que se conecta a las calculadoras de bolsillo permite una gran rapidez en la búsqueda de los programas y sus datos.

```
01♦LBL "TERN-P"
02 FIX 0
03 CF 29
04 1,10001
05 STO 00
06♦LBL 05
07 RCL 00
08 INT
09 X↑2
10 STO 02
11 RCL 00
12 1
13 +
14 STO 01
15♦LBL 06
16 RCL 01
17 INT
18 X↑2
19 STO 03
20 RCL 02
21 RCL 03
22 +
23 STO 04
24 SQRT
25 STO 05
26 INT
27 X↑2
28 RCL 04
29 X=Y?
30 XEQ 08
31 ISG 01
32 GTO 06
33 ISG 00
34 GTO 05
35 BEEP
36 STOP
37♦LBL 07
38 CLA
39 ARCL 00
40 "├+"
41 ARCL 01
42 "├="
43 ARCL 05
44 AVIEW
45 RTN
46♦LBL 08
47 RCL 01
48 INT
49 STO 07
50 RCL 00
51 INT
52 STO 08
53♦LBL 03
54 RCL 07
55 RCL 08
56 /
57 INT
58 RCL 08
59 *
60 CHS
61 RCL 07
62 +
63 STO 06
64 X=0?
65 GTO 09
66 RCL 08
67 STO 07
68 RCL 06
69 STO 08
70 GTO 03
71♦LBL 09
72 RCL 08
73 1
74 X=Y?
75 XEQ 07
76 RTN
77 END
```

```
3+4=5
5+12=13
7+24=25
8+15=17
9+40=41
11+60=61
12+35=37
13+84=85
16+63=65
20+21=29✓
20+99=101
28+45=53
33+56=65
36+77=85
39+80=89
48+55=73
60+91=109
65+72=97
```

Figura 53. Listado del programa n.º 25.

Mejor resultado se obtiene con el programa n.º 25 de la figura 53, adecuado para la calculadora HP-41C, que tiene el resultado de las 18 ternas obtenidas junto al listado, en las que no se ha indicado el cuadrado por dificultades de la impresora.

En vez de la anterior DSZ, se utiliza la análoga ISG pero en dos memorias distintas, en las líneas 31 y 33. Estas memorias se inician con el valor de control de la línea 04 que indica que se iniciará la cuenta con el valor de la parte entera, el 1, y se llegará hasta 100 con incrementos de 01. Este sistema permite una gran variedad de aplicaciones puesto que quedan determinados los tres parámetros del bucle repetitivo: el principio, el final y el incremento.

El resultado se imprime con el LBL 07 de la línea 37. El LBL 08 sirve para cambiar los valores de las memorias antes de pasar al Algoritmo de Euclides, en el LBL 03.

```
  5 PRINT "T.PITAG.
    "
 10 FOR X=1 TO 100
 20 FOR Y=X+1 TO 10
    0
 30 A=X*X+Y*Y:D=SQR
     A
 40 IF INT D*INT D=
    A THEN 100
 50 NEXT Y
 60 NEXT X
 70 END

100 B=X:C=Y
110 E=INT (B/C):R=B
    -C*E
120 IF R=0 THEN 150
130 B=C:C=R:GOTO 11
    0
150 IF C=1 THEN 200
160 GOTO 50
200 PRINT X;" + ";Y
    ;" = ";D
210 GOTO 50
```

Figura 54. Listado del programa n.º 26.

Y análogo resultado se obtiene con el programa n.º 26 de la figura 54 para la calculadora PB-100. Los dos bucles FOR...NEXT están imbricados o uno dentro del otro. En la línea 100 hay el cambio de variables para poder utilizar el algoritmo de Euclides que está entre las líneas 110 y 130, mientras que el resultado se presenta o imprime según la instrucción de la línea 200 y se termina la fase con la instrucción de la línea 210.

Los programas resueltos con la estructura iterativa tienen una mayor complejidad, pero a la vez permiten solucionar los más complejos problemas dentro de la capacidad de la memoria RAM de las calculadoras de bolsillo.

DIVERSAS POSIBILIDADES DE LAS CALCULADORAS PROGRAMABLES

Sea por los periféricos a que se pueden conectar o por las funciones o comandos de su unidad central, las calculadoras programables tienen un sinfin de posibilidades dentro de la capacidad de su reducida memoria. En este apartado se comentarán algunos programas que muestran las posibilidades del módulo de tiempo, del direccionamiento indirecto, de la capacidad de producir gráficos, y otras que se podrían citar pero la falta de espacio limita su descripción.

El módulo de tiempo que tienen algunas calculadoras, ya sea adaptable o perteneciente al circuito de la unidad central, permite una gran cantidad de aplicaciones. El de la calculadora HP-41C consta de un reloj digital controlado por un cristal de cuarzo, que no sólo proporciona la hora a la centésima de segundo, con posibilidad de imprimirla, sino que también tiene asociadas las funciones de calendario, las cuales permiten hacer cálculos por programa con referencia a los días transcurridos entre fechas tan distantes desde el año 1900 hasta el 2199. Además permite pasar a las memorias sucesivas la hora en que se ha pulsado la tecla ENTER, muy útil para la toma de distintos tiempos, y puede usarse como reloj avisador que a una hora y día determinados da una señal sonora a la vez que por la pantalla se visualiza el mensaje previamente programado.

Como aplicación de este módulo y del direccionamiento indirecto que se comentará también para el programa siguiente, veamos en el programa n.º 27 de la figura 55 una modificación del programa n.º 14 que determina el mayor número aleatorio generado.

Puede verse por el resultado que acompaña a este programa, que después de entrarle la semilla va imprimiendo los mayores números aleatorios que genera, a la vez que los acompaña del tiempo en minutos que han tardado en obtenerse. Se ha detenido el programa a los 60 minutos de su funcionamiento, y por los números aleatorios obtenidos y

su tiempo no se observa ninguna regularidad y sí una buena dispersión, tal como tiene que ser.

El generador es distinto del programado anteriormente, aunque es del mismo tipo. Está al principio del LBL 05 y en vez del número *pi* utiliza la cifra decimal 0,211327, que

```
01♦LBL "ALEAT"
02 "SEMILLA="
03 PROMPT
04 STO 00
05 0
06 STO 01
07 4
08 STO 02
09 TIME
10 HR
11 60
12 *
13 STO 03
14♦LBL 05
15 RCL 00
16 9821
17 *
18 ,211327
19 +
20 FRC
21 STO 00
22 RCL 01
23 X>Y?
24 GTO 05
25 RCL 00
26 STO 01
27 STO IND 02
28 1
29 ST+ 02
30 FIX 9
31 RCL 01
32 PRX
33 FIX 2
34 CLA
35 TIME
36 HR
37 60
38 *
39 RCL 03
40 -
41 ARCL X
42 ACA
43 "=MINUTOS"
44 ACA
45 ADV
46 GTO 05
47 END
```

```
          SEMILLA=
                  0,1717    RUN
          0,477027000       ***
                  0,03=MINUTOS
          0,775048000       ***
                  0,09=MINUTOS
          0,957735000       ***
                  0,13=MINUTOS
          0,986864000       ***
                  0,25=MINUTOS
          0,989093000       ***
                  2,01=MINUTOS
          0,991285000       ***
                  2,25=MINUTOS
          0,993126000       ***
                  2,59=MINUTOS
          0,997964000       ***
                  4,44=MINUTOS
          0,998012000       ***
                 17,34=MINUTOS
          0,998695000       ***
                 17,76=MINUTOS
          0,998919000       ***
                 31,32=MINUTOS
          0,999692000       ***
                 32,56=MINUTOS
          0,999767000       ***
                 33,34=MINUTOS
          0,999774000       ***
                 33,65=MINUTOS
          0,999924000       ***
                 46,64=MINUTOS
```

Figura 55. Listado del programa n.º 27.

sumada al producto del número anteriormente generado por 9821, queda una parte fraccionaria que es el nuevo número aleatorio.

Para la obtención del tiempo de aparición, como condiciones iniciales se obtiene la hora por el comando TIME en la línea 09, hora que pasa de minutos y segundos a decimal de hora por HR, y a minutos y centésimas al multiplicarla por 60. Luego se almacena en la memoria 03 para que cuando se vuelva a obtener en la línea 35 se puede hallar la diferencia de la nueva hora con la primera, línea 40, que es la que se imprime seguida de la palabra «=MINUTOS».

Por el resultado puede comprobarse que es un buen generador aleatorio (figura en el manual de la calculadora HP-41C) ya que después de media hora de generar números distintos no entra en un ciclo repetitivo, como puede verse por el resultado obtenido a 46 minutos de funcionamiento.

Otro aspecto interesante de este programa es el direccionamiento indirecto de la línea 27, por el que se almacenan los números aleatorios mayores en distintas memorias a partir de la 04.

Por el direccionamiento indirecto se va a una dirección relativa que está indicada por el resultado de una fórmula o por el contenido de una memoria.

En este programa el número que tenemos en la memoria 01 se almacena en la memoria cuyo número está indicado por el contenido de la memoria 02, la cual empieza con el valor 4 y aumenta de uno en uno cada vez que se encuentra un número aleatorio mayor. Puede verse este incremento en las líneas 28 y 29 del programa.

Las memorias o registros utilizados en este programa son: R_{00}=Auxiliar del generador aleatorio. R_{01}=Valor máximo generado. R_{02}=Contador del registro indirecto. R_{03}=Auxiliar para el origen del tiempo. R_{04}=Primer valor máximo generado. R_{05}=Segundo valor máximo generado; y así sucesivamente, por lo que hay que utilizar un SIZE suficientemente grande para tener memorias o registros vacíos, en los cuales estarán los valores máximos y se podrán consultar si no se dispone de impresora.

El direccionamiento indirecto es en este programa a efectos de almacenamiento, pero puede también utilizarse con GOTO, GOSUB y otras direcciones, es decir: si el programa encuentra GTO IND 02 se irá al LBL que le indica el contenido de la memoria 02. Es más difícil este direccio-

```
  5 PRINT "DISCURSO
     AL AZAR"
 10 PRINT "CAMARADA
    S:"
 20 GOSUB 700:GOSUB
     (A*10+100)
 30 GOSUB 700:GOSUB
     A*10+200
 40 GOSUB 700:GOSUB
     A*10+300
 50 GOSUB 700:GOSUB
     A*10+400
 60 GOSUB 700:GOSUB
     A*10+500
 70 GOTO 20
110 PRINT "EL AFAN
    ESTRUCTURADOR D
    E LA ";
115 PRINT "ADMINIST
    RACION, ";:RETU
    RN
120 PRINT "LA REALI
    ZACION DE LAS E
    TAPAS ";:RETURN

130 PRINT "LA PRUDE
    NCIA EN LA EJEC
    UCION ";:RETURN

140 PRINT "EL IDEAR
    IO DEL TRABAJO
    ";:RETURN
210 PRINT "SOBREPAS
    A ADMIRABLEMENT
    E ";:RETURN
220 PRINT "IMPLICA
    ";:RETURN
230 PRINT "SIRVE A
    ";:RETURN
240 PRINT "CONSIGUE
    ";:RETURN
```

```
310 PRINT "LA FUNCI
    ON ORGANIZADORA
     DE ";
315 PRINT "LOS DEBE
    RES FUNDAMENTAL
    ES ";:RETURN
320 PRINT "LA CREAC
    ION DE LAS LEYE
    S ";:RETURN
330 PRINT "LA PREPA
    RACION DE LAS T
    AREAS ";:RETURN

340 PRINT "LA EJECU
    CION DE LAS FAS
    ES ";:RETURN
410 PRINT "DE LOS C
    ARGOS DEL SERVI
    CIO ";:RETURN
420 PRINT "DE LA FO
    RMACION HUMANA
    ";:RETURN
430 PRINT "DE LA AR
    GUMENTACION SOL
    IDA ";
435 PRINT "DEL PROC
    ESO ACTUAL ";:R
    ETURN
440 PRINT "DE LA EC
    ONOMIA MODERNA
    ";:RETURN
510 PRINT "MAS, ";:
    RETURN
520 PRINT "DE IGUAL
     MANERA, ";:RET
    URN
530 PRINT "YA QUE,"
    ;:RETURN
540 PRINT "ASI COMO
    ,";:RETURN
700 A=INT (RAN#*4+1
    ):RETURN
```

```
READY P0

DISCURSO AL AZAR
CAMARADAS:
LA PRUDENCIA EN LA E
JECUCION SOBREPASA A
DMIRABLEMENTE LA PRE
PARACION DE LAS TARE
AS DE LA ECONOMIA MO
DERNA ASI COMO,LA RE
ALIZACION DE LAS ETA
PAS SIRVE A LA EJECU
CION DE LAS FASES DE
 LA ARGUMENTACION SO
LIDA DEL PROCESO ACT
UAL YA QUE,LA REALIZ
ACION DE LAS ETAPAS
CONSIGUE LA EJECUCIO
N DE LAS FASES DE LA
 FORMACION HUMANA DE
 IGUAL MANERA, EL ID
EARIO DEL TRABAJO CO
NSIGUE LA EJECUCION
DE LAS FASES DE LA A
RGUMENTACION SOLIDA
DEL PROCESO ACTUAL M
AS, LA REALIZACION D
E LAS ETAPAS CONSIGU
E LA EJECUCION DE LA
S FASES DE LA FORMAC
ION HUMANA MAS, EL A
FAN ESTRUCTURADOR DE
 LA ADMINISTRACION,
SOBREPASA ADMIRABLEM
ENTE LA PREPARACION
```

Figura 56. Listado del programa n.º 28.

namiento, pero simplifica mucho los programas que con pocas líneas obtienen una gran variedad de efectos.

Esto lo podemos comprobar para la calculadora PB-100 en el programa n.º 28 de la figura 56, que obtiene las frases de un discurso obtenidas al azar, tal como puede comprobarse por el resultado que se adjunta con el listado.

Para ello se utiliza el generador aleatorio que posee la propia calculadora con la función RAN, el cual puede verse que en la línea 700 se multiplica por 4 para tener los resultados 1, 2, 3 y 4 al azar.



Calculadora de bolsillo conectada a un multímetro para el trabajo en un laboratorio de diseño de circuitos integrados.

Se seleccionan secuencialmente cinco partes de la sintaxis de una frase y de cada una de ellas se han hecho cuatro versiones distintas que se obtienen al azar. La selección secuencial queda en pocos pasos de programa por el direccionamiento indirecto y figura en las líneas de la 20 a la 60, en las que primero se obtiene un número del 1 al 4, por la subrutina citada de la línea 700 y luego se va a una línea de forma indirecta, como puede verse en la línea 20 que se ha puesto entre paréntesis.

Se ha preferido ir a una línea del programa porque con el comando PRINT se pueden imprimir hasta 30 caracteres, mientras que si se hubiese ido al contenido de variables sólo pueden contener 7 caracteres en esta calculadora y se emplearía demasiada memoria RAM. Cuando estos caracteres no caben en una línea se pasa a la siguiente a través del correspondiente punto y coma, y por RETURN se vuelve a la secuencia siguiente.

